

Research Article

An Algorithm for Finding Functional Modules and Protein Complexes in Protein-Protein Interaction Networks

Guangyu Cui,¹ Yu Chen,¹ De-Shuang Huang,² and Kyungsook Han¹

¹ School of Computer Science and Engineering, Inha University, Incheon 402-751, South Korea

² Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, China

Correspondence should be addressed to Kyungsook Han, khan@inha.ac.kr

Received 1 September 2007; Revised 12 November 2007; Accepted 24 December 2007

Recommended by Daniel Howard

Biological processes are often performed by a group of proteins rather than by individual proteins, and proteins in a same biological group form a densely connected subgraph in a protein-protein interaction network. Therefore, finding a densely connected subgraph provides useful information to predict the function or protein complex of uncharacterized proteins in the highly connected subgraph. We have developed an efficient algorithm and program for finding cliques and near-cliques in a protein-protein interaction network. Analysis of the interaction network of yeast proteins using the algorithm demonstrates that 59% of the near-cliques identified by our algorithm have at least one function shared by all the proteins within a near-clique, and that 56% of the near-cliques show a good agreement with the experimentally determined protein complexes catalogued in MIPS.

Copyright © 2008 Guangyu Cui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Proteins in a highly connected subgraph of a protein interaction network usually share a common function [1]. Therefore, a highly connected subgraph such as clique and near-clique in a protein interaction network can be used to predict the function of uncharacterized proteins in the highly connected subgraph. Finding a clique with a maximum size in a graph is an NP-hard problem [2]. There are several heuristic algorithms for the maximum clique problem [2, 3], but most of them focus on finding a complete subgraph (i.e., clique) and cannot be used to find near-cliques.

Several topological analysis methods have been developed for identifying biologically meaningful groups from protein interaction networks or for assessing the reliability of protein interactions. A recent program called CFinder [4, 5] finds overlapping cliques in protein interaction networks. It allows a protein to belong to more than one clique, but cannot find near-cliques. Our study shows that the near-cliques can reveal higher functional coherence than the overlapping cliques.

The primary focus of this study is to find functional groups by identifying cliques and near-cliques in protein interaction networks. This study attempts to answer two ques-

tions as follows. “Can we efficiently find all cliques and near-cliques?” and “does a dense subgraph such as clique and near-clique indeed represent a functional module or protein complex?” This study demonstrates that the answers to both questions are “yes.” This paper presents an algorithm for finding near-cliques and its application to the interaction network of yeast proteins.

2. ALGORITHMS FOR FINDING NEAR-CLIQUE

A clique is a complete graph $G = (N, E)$ in which every node is connected to every other node in the graph. In our previous work, we developed a heuristic algorithm and implemented the algorithm in a program called InterViewer [6], which identifies all edge-disjoint cliques (i.e., cliques that do not share an edge).

Our experience with protein interaction networks suggests that a near-clique as well as a clique often represents a biologically meaningful unit such as functional module or protein complex. A near-clique is almost a clique but is not a clique due to a few missing edges. We consider near-cliques of the following *basic* types, which are biologically meaningful clusters (see Figure 1).

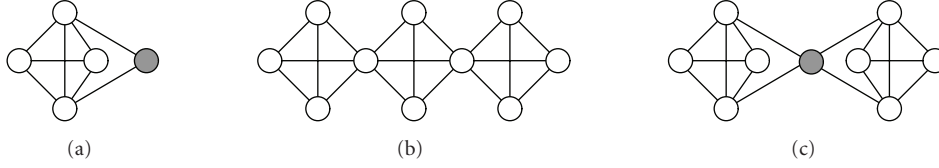


FIGURE 1: Near-cliques of types A, B, and C. Proteins outside a clique are represented as shaded nodes.

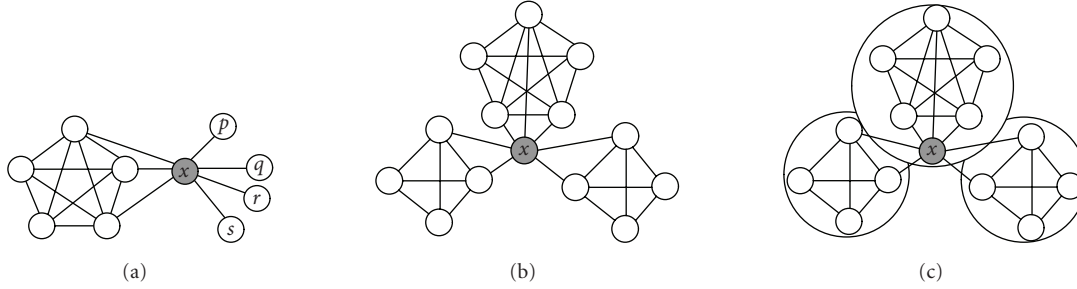


FIGURE 2: (a) After removing nodes p , q , r , and s and their edges, node x forms a near-clique of type A with the remaining nodes. (b) This graph becomes a near-clique G of type C since $\text{indegree}(x, G) \geq 0.5|G|$. (c) A big near-clique is too big (e.g., near-clique with more than 50 nodes) and is split into smaller near-cliques (in this example, 3 small near-cliques).

Type A

When a protein outside a clique interacts with two or more proteins in the clique, the protein and the clique forms a near-clique.

Type B

When a clique shares a protein with other cliques, the cliques form a near-clique.

Type C

When two or more cliques interact with a common protein outside them and the protein has at least two interactions with each clique, the cliques and the protein form a near-clique.

The near-cliques of types A and C can be refined using the indegree and outdegree of a node (there is no change to the near-clique of type B). For a node x in subgraph $G' \subset G$, $\text{indegree}(x, G')$ is the number of the edges connecting node x to other nodes in G' , and $\text{outdegree}(x, G')$ is the number of edges connecting node x to other nodes that are in G but not in G' . We use the definition of a community in a strong sense [7] to find more near-cliques in a graph.

Definition 1. A subgraph G' is a community in a strong sense if $\text{indegree}(x, G') > \text{outdegree}(x, G')$ for every x in G' .

The original definition of a strong community misses many near-cliques due to a single node in the communities. For example, in Figure 2(a), node x cannot belong to a near-clique since $\text{indegree}(x, G') = 3 < \text{outdegree}(x, G') = 4$. Likewise, node x in Figure 2(b) cannot belong to a near-clique because $\text{indegree}(x, G') < \text{outdegree}(x, G')$. Thus, nodes with only one edge connected to them and their edges are removed from the graph when we search near-cliques in the graph. In the graph of Figure 2(a), nodes p , q , r , and s

and their edges are removed. After removing them, node x and the existing clique form a near-clique of type A. A cluster that satisfies $\text{indegree}(x, G') \geq 0.5|G'|$ for every x in G' , where $|G'|$ is the number of nodes in G' , forms a near-clique, too. The example shown in Figure 2(b) becomes a near-clique since it satisfies $\text{indegree}(x, G') \geq 0.5|G'|$ even if it does not satisfy $\text{indegree}(x, G') < \text{outdegree}(x, G')$.

Therefore, a near-clique G of basic types A and C should satisfy at least one of the following conditions.

- (1) $\text{indegree}(x, G) \geq \text{outdegree}(x, G)$ for every x in G .
- (2) $\text{indegree}(x, G) \geq 0.5|G|$.

After finding all edge-disjoint cliques first, we identify near-cliques as follows. More detailed description of finding near-cliques are outlined in Algorithms 1 and 2. In the algorithms, cldx represents the index of a clique.

- (1) Assign every node of a clique the index of the clique containing the node.
- (2) When a node of a clique has already an assigned clique index, assign the index to all nodes of the clique, and merge two cliques into a near-clique of type B.
- (3) When a node x outside a clique forms a basic near-clique G of type A due to the interactions with two or more proteins in the clique, and either $\text{indegree}(x, G) \geq \text{outdegree}(x, G)$ or $\text{indegree}(x, G) \geq 0.5|G|$ is true, assign the index of the clique to the node.
- (4) When two or more cliques form a near-clique G due to two or more interactions with a common protein outside the cliques, and either $\text{indegree}(x, G) \geq \text{outdegree}(x, G)$ or $\text{indegree}(x, G) \geq 0.5|G|$ is true, merge the cliques and the protein into a near-clique of type C. A near-clique is formed by selecting nodes with the same clique index (cldx) as those nodes with $\text{cldx} > 0$.

```

(1) for all node  $N \in G$  do
(2)    $N.cIdx = 0$                                      {initialize  $cIdx$  of all nodes to 0}
(3) end for
(4)  $curCIdx = 1$                                        {set the current clique index to 1}
(5) for all node  $N \in G$  do
(6)   if (isClique( $N$ )) then {if the node  $N$  belongs to a clique}
(7)     for all edge  $E \in N$  do
(8)       if ( $E.target.cIdx > 0$ ) then {if the  $cIdx$  of the node connected to  $N$  is positive}
(9)         for all  $tmpN \in G$  do {for all nodes in  $G$ }
(10)          if ( $tmpN.cIdx = E.target.cIdx$ ) then
(11)             $tmpN.cIdx = curCIdx$                    {assign  $curCIdx$  to  $tmpN$  as its  $cIdx$ }
(12)          end if
(13)        end for
(14)      else
(15)         $E.target.cIdx = curCIdx$ 
(16)      end if
(17)    end for
(18)     $N.cIdx = curCIdx$                                {set  $cIdx$  of  $N$  to  $curCIdx$ }
(19)     $curCIdx++$                                        {increase  $curCIdx$  by one}
(20)  end if
(21) end for

```

ALGORITHM 1: AssignNearCliqueIdx.

```

(1) for all node  $N \in G$  do
(2)   if ( $N.cIdx = 0$ ) then                             {find node outside the clique}
(3)      $qCliqueCnts = \emptyset$  { $qCliqueCnts$  the number of edges, which the node  $N$  connected with different near-cliques}
(4)     for all edge  $E \in N$  do
(5)       if ( $E.target.cIdx > 0$ ) then
(6)          $qCliqueCnts[E.target.cIdx]++$ 
(7)       end if
(8)     end for
(9)      $qC\text{ value} = 0$                                   {initialize  $cIdx$  of node  $N$ }
(10)    for all ( $c \in qCliqueCnts$ ) do
(11)      if ( $(c > 1)$  and  $\text{indegree}(x, G') \geq \text{outdegree}(x, G')$ ) or ( $(c > 1)$  and  $\text{indegree}(x, G') \geq 0.5 * |G'|$ ) then {a node outside a clique interacts with multiple nodes in the clique, and either  $\text{indegree}(x, G') \geq \text{outdegree}(x, G')$  or  $\text{indegree}(x, G') \geq 0.5 * |G'|$  is true}
(12)        if ( $qC\text{ value} > 0$ ) then
(13)          for all  $tmpN \in G$  do
(14)            if ( $tmpN.cIdx = qC\text{ value}$ ) then
(15)               $tmpN.cIdx = qC\text{ value}$                  {near-clique of type C}
(16)            end if
(17)          end for
(18)        else
(19)           $qC\text{ value} = c$                              {near-clique of type A}
(20)        end if
(21)      end if
(22)    end for
(23)     $N.cIdx = qC\text{ value}$                                {assign  $qC$  value to node  $N$  as its  $cIdx$ }
(24)  end if
(25) end for

```

ALGORITHM 2: ExtendNearClique.

Since the most relevant processes form a group of proteins of moderate size in biological networks [8], we obtain near-cliques smaller than the maximum size specified by a user. That is, when a near-clique bigger than the maximum size is found (e.g., near-clique with more than 50 nodes), it is

split into smaller near-cliques (3 near-cliques in Figure 2(c)). The way we split a big near-clique is as follows. When our program finds a big near-clique with the minimum clique size set to k , we rerun the program on the big near-clique with the minimum clique size set to $k + 1$ to find a new

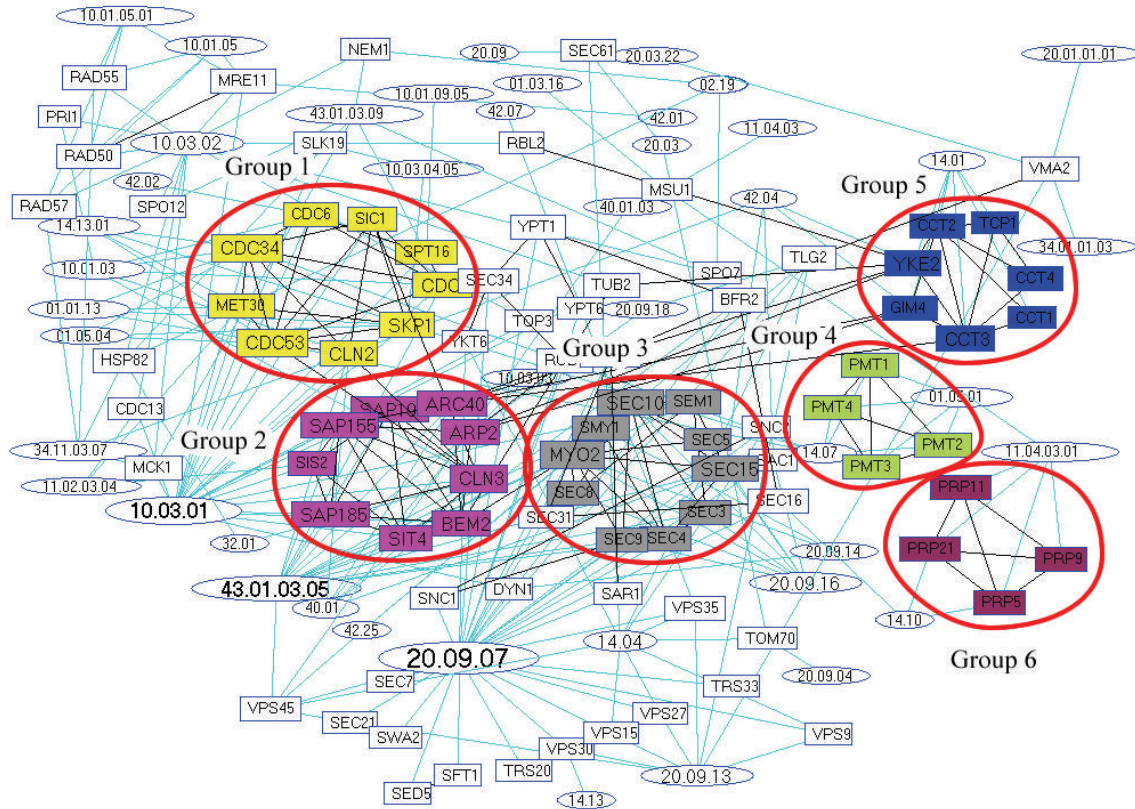


FIGURE 3: Six near-cliques found in yeast protein interaction networks. Proteins in each near-clique share at least one function with other proteins within the near-clique.

clique and a near-clique with the clique. After removing the new near-clique from the original, big near-clique, we run the program again with the minimum clique size set to k . The big near-clique shown in Figure 2(c) is split into 3 small near-cliques with at least 4 proteins each.

3. RESULTS AND COMPARISON WITH EXPERIMENTAL DATA

We tested the algorithms on the data with 8,397 interactions between 4,380 yeast proteins, which is the combined data of Ito et al. [9], Uetz et al. [10], and MIPS (<http://mips.gsf.de>) with redundant data removed. To every protein in the near-cliques, we assigned the functional categories of the Functional Catalog (FunCat) version 2.0 [11], which includes 97 functional categories. There are six levels of hierarchy in the FunCat structure.

In the data with 8,397 interactions between 4,380 yeast proteins, we found 100 near-cliques with the minimum size of a clique set to 3 and the maximum size of a near-clique set to 40. Only one near-clique contains more than 40 proteins, and so it was split into 17 small near-cliques, resulting in total 116 near-cliques. Figure 3 shows an example of the network of yeast protein interactions with 6 near-cliques. Proteins in each near-clique share at least one function with other proteins within the near-clique.

As shown in Table 1, 68 (59%) out of the 116 near-cliques have at least one function shared by all the proteins in the near-cliques (100% sharing), and 39 near-cliques have a function shared by more than 50% of the proteins in the near-cliques, supporting data are available at <http://wilab.inha.ac.kr/ppi/homepage.mht>. Only 9 near-cliques have no function shared by >50% of the proteins in the near-cliques. As shown in Figure 4, the functional coherence of each near-clique is high. The functional coherence was computed by the ratios of the number of proteins having a specific functional category to the group size (i.e., the number of proteins in the group).

Interestingly, most near-cliques found by our algorithm belong to multifunctional categories. For example, two functional categories are common to all the proteins in a near-clique of Figure 5. As shown in Table 2, the near-clique identified as group 93 by our program is involved in both stress response (functional category 32.01) and biosynthesis of vitamins, cofactors, and prosthetic groups (functional category 01.07.01).

Near-cliques may correspond to protein complexes in addition to functional modules. So, we compared the near-cliques identified by our algorithms with known yeast protein complexes, which are cataloged in the MIPS *Saccharomyces cerevisiae* genome database (<http://mips.gsf.de/genre/proj/yeast>). For each near-clique, we found a best-matching protein complex by minimizing

TABLE 1: Functional groups identified from the yeast protein interaction data. 68 modules have at least one function shared by all the proteins in the groups (100% sharing), and 39 groups have a function shared by more than 50% of the proteins in the groups. Only 9 groups have no function shared by >50% of the proteins in the group. This table shows only one function with the highest functional coherence in each group. All the functions shared by more than 50% of the proteins in each group are available at <http://wilab.inha.ac.kr/ppi/homepage.mht>.

Group ID	Proteins in the group	Common function (proportion of proteins with the function)	Group ID	Proteins in the group	Common function (proportion of proteins with the function)	Group ID	Proteins in the group	Common function (proportion of proteins with the function)
1	3	11.02.02 (100%)	40	3	10.01.03.03 (100%)	79	3	42.01 (100%)
2	5	14.01 (100%)	41	3	20.09.13 (100%)	80	3	30.01.05.01 (66.7%)
3	4	42.10 (100%)	42	5	42.01 (80%)	81	3	42.04.03 (100%)
4	12	42.10.05 (75%)	43	5	40.01 (80%)	82	4	42.25 (100%)
5	4	01.03.16.01 (75%)	44	3	14.10 (100%)	83	3	none
6	6	11.02.03.04 (100%)	45	8	12.04.01 (87.5%)	84	3	01.06.01.07.11 (100%)
7	4	14.07.02.01 (100%)	46	4	14.13.01.01 (100%)	85	3	32.01.07 (66.7%)
8	6	10.03.01 (100%)	47	4	10.01.05.01 (100%)	86	4	34.11.03.07 (100%)
9	22	11.04.01 (63.6%)	48	3	10.01.05.01 (100%)	87	3	20.09.07 (100%)
10	21	20.09 (66.7%)	49	6	20.09.04 (100%)	88	3	02.19 (100%)
11	3	none	50	4	32.01 (100%)	89	3	16.19.03 (100%)
12	8	11.04.03.05 (100%)	51	3	10.01.03 (100%)	90	4	2.07 (75%)
13	11	10.03.01 (63.6%)	52	3	12.04.03 (66.7%)	91	3	16.03.01 (100%)
14	7	10.03.01 (76.5%)	53	13	20.09.07.03 (61.5%)	92	3	10.01.05.01 (100%)
15	4	1.03 (50%)	54	8	11.02.03.01 (100%)	93	7	32.01 (100%)
16	5	01.05.01.03.02.02 (100%)	55	4	20.09.07.03 (100%)	94	3	40.20 (66.7%)
17	3	16.03.01 (100%)	56	5	none	95	3	34.01.01.03 (100%)
18	4	11.04.02 (100%)	57	5	20.09.01 (100%)	96	4	43.01.03.05 (100%)
19	5	40.01 (80%)	58	5	20.09.18 (80%)	97	3	14.04 (100%)
20	3	18.02.01 (60%)	59	5	01.04.01 (80%)	98	4	20.09.13 (100%)
21	23	43.01.03.05 (82.6%)	60	3	43.01.03.05 (100%)	99	3	16.03.01 (66.7%)
22	4	32.01 (50%)	61	5	11.04.01 (100%)	100	12	43.01.03.05 (91.7%)
23	4	11.02.03.04.01 (100%)	62	5	20.09.10 (100%)	101	6	10.03.01.01.03 (100%)
24	4	14.13.01.01 (100%)	63	3	43.01.03.09 (66.7%)	102	9	16.01 (88.9%)
25	36	none	64	3	11.02.03.04 (100%)	103	7	43.01.03.05 (100%)

TABLE 1: Continued.

Group ID	Proteins in the group	Common function (proportion of proteins with the function)	Group ID	Proteins in the group	Common function (proportion of proteins with the function)	Group ID	Proteins in the group	Common function (proportion of proteins with the function)
26	4	20.09.07.03 (100%)	65	3	34.11.03.13 (100%)	104	5	43.01.03.05 (80%)
27	10	42.04 (50%)	66	5	16.19.03 (80%)	105	11	20.09.07.03 (100%)
28	11	14.13.01.01 (100%)	67	4	11.04 (100%)	106	7	10.03.04.03 (85.7%)
29	6	43.01.03.05 (83.3%)	68	3	10.01.09.05 (66.7%)	107	6	40.01 (50%)
30	8	12.04 (100%)	69	4	01.04.01 (100%)	108	9	10.03.01.01 (88.9%)
31	4	10.03.01 (100%)	70	3	11.06.01 (100%)	109	3	20.09.07.27 (100%)
32	4	12.04.02 (75%)	71	6	none	110	3	20.09.14 (100%)
33	3	34.01.01.01 (100%)	72	3	20.09.13 (100%)	111	12	43.01.03.05 (100%)
34	5	none	73	3	11.02.03.04 (100%)	112	5	10.03.04.05 (100%)
35	3	11.04.01 (66.7%)	74	3	42.10.03 (100%)	113	5	10.03.04.05 (100%)
36	31	11.02.03.04 (80.6%)	75	3	none	114	5	42.10.03 (80%)
37	4	16.03.01 (100%)	76	7	20.09.04 (100%)	115	3	none
38	6	43.01.03.05 (100%)	77	3	none	116	5	43.01.03.05 (80%)
39	7	14.04 (57.1%)	78	3	10.03.02 (100%)	—	—	—

TABLE 2: Functional annotation of group 93 shown in Figure 5. The code represents functional category.

Group	Protein	Protein functional categories	
Group 93	YFL059w	32.01	01.07.01
	YMR096w	32.01	01.07.01
	YMR322c	32.01.07	01.07
	YNL334c	32.01	01.07.01
	YMR095c	32.01	01.07.01
	YNL333w	32.01	01.07.01
	YFL060c	32.01	01.07.01

the probability of a random overlap between the two, using the following equation [4, 5]:

$$P_{\text{overlap}} = \frac{\binom{n_2}{k} \binom{N-n_2}{n_1-k}}{\binom{N}{n_1}}, \quad (1)$$

where n_1 , n_2 are the sizes of a known protein complex and a computed module, k is the number of their common proteins, and N is the size of the network.

As shown in Table 3, 65 near-cliques (56% of the total 116 near-cliques) identified by our algorithm show a good agreement ($\ln(P_{\text{overlap}}) < -14$) with the protein complexes cataloged in MIPS.

To compare the functional coherence of the groups found by our program with that of cliques found by CFinder, we tested both programs on the same dataset. 75.9% of the groups identified by our program have at least two functional categories shared by all the proteins in the groups, whereas 63.1% of the groups identified by CFinder have at least two functional categories shared by all the proteins in the groups (Table 4). This result indicates that our program finds groups with stronger functional coherence than CFinder.

Table 5 shows the actual running times of our program and CFinder on three datasets of yeast protein interactions. Our program is faster than CFinder on all datasets, and the difference in speed becomes more obvious as the dataset becomes bigger.

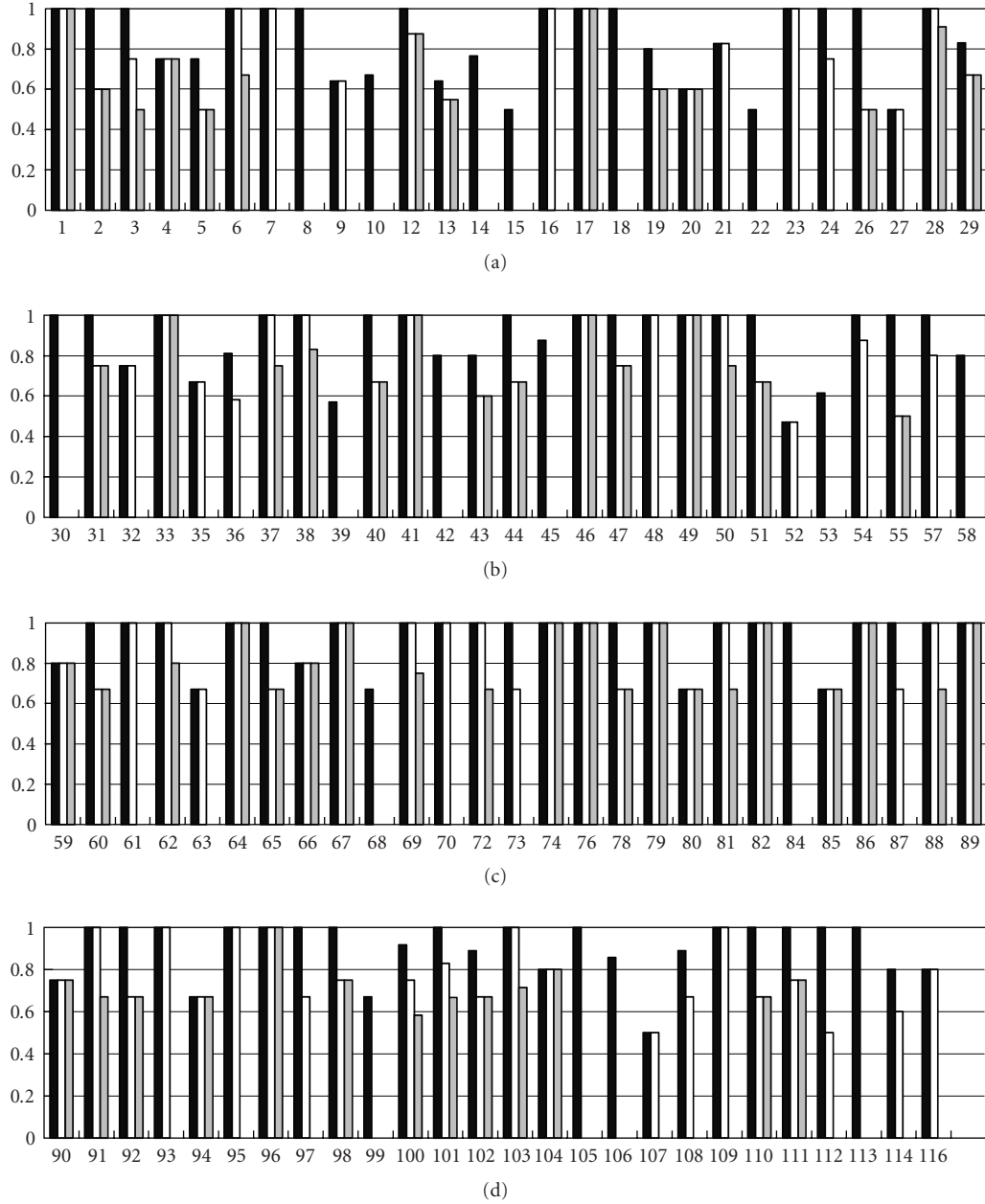


FIGURE 4: The functional coherence in each of the 116 groups, computed as the ratio of the number of proteins having a specific functional category to the number of proteins in the group. The black, white, and grey bars represent functional categories with the ratios ≥ 0.5 and the maximum number of such ratios is limited to 3 in each group.

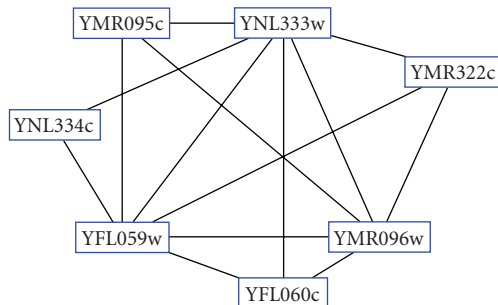


FIGURE 5: Group 93 identified as a near-clique by our algorithm.

4. CONCLUSION

Identifying hidden topological structures of protein interaction networks often unveil biologically relevant functional groups and structural complexes. We developed an efficient heuristic algorithm for finding cliques and near-cliques in protein interaction networks. From the interaction data of yeast proteins, the algorithm identified 116 near-cliques. Comparison with the experimental data showed that 59% of the near-cliques have at least one function shared by all the proteins within a near-clique, and that 56% of the near-cliques show a good agreement with known protein

TABLE 3: The near-cliques matched with experimentally determined protein complexes cataloged in MIPS. The overlap column represents the number of proteins common to the near-cliques and the protein complexes.

Group ID	Group size	MIPS tag of protein complex	Protein complex size	Overlap (common proteins)	$\ln(P_{\text{overlap}})$	Main functional categories
3	4	295	2	2	-14.28	42.10/43.01.03.09
5	4	510.190.110	13	3	-16.32	01.03.16.01
6	6	320	8	4	-23.40	11.02.03.04/10.01.09.05/16.03.01
9	22	550.1.149	88	21	-84.60	11.04.01/12.01
12	8	550.1.148	35	8	-39.49	11.04.03.05/16.03.03/11.04.03.01
13	11	550.1.7	10	8	-47.55	10.03.01/14.01/16.01
14	17	140.30	32	11	-46.65	10.03.01
16	4	550.1.44	9	3	-17.55	14.07.02.02/01.05.01.03.02.02
17	3	410.40.20	3	3	-23.36	16.03.01 /16.03.01 /10.01.05.01/10.01.03.05 /10.01.03.01
18	4	440.30.30	11	4	-24.56	11.04.02
21	23	470.20	5	5	-26.71	43.01.03.05/34.11.03.07
23	4	510.160	4	4	-30.36	11.02.03.04.01/01.05.04
24	4	360.10	36	4	-19.38	14.13.01.01/14.07.11
25	36	550.1.138	36	11	-34.43	none
26	4	550.2.317	3	3	-21.98	20.09.07.03/20.09.07.05/14.10
27	10	130	8	4	-20.77	42.04/16.01
28	11	60	11	11	-74.71	14.13.01.01/10.03.01.01.11/14.07.05 /14.10 /16.01 /16.19.03
29	6	120.20	4	4	-27.65	43.01.03.05/40.01/34.07.01/34.01/32.01.09 /11.02.02/10.03.01.01.09/10.03.01.01.03
30	8	550.1.142	25	4	-16.68	12.04
31	4	140.30.30.30	3	2	-13.18	10.03.01/42.04/20.09
32	3	510.20	4	3	-21.98	12.04.02/12.01.01
36	31	230.20.20	16	14	-67.99	11.02.03.04/10.01.09.05
37	4	410.30	16	4	-22.85	16.03.01/10.01.03.03/10.01.03.01/16.19.03
38	6	550.1.81	7	5	-32.29	43.01.03.05/20.09.16.09.03/16.01/20.09.07.27 /10.03.03
40	3	410.30	16	3	-17.03	10.01.03.03/10.01.09.05 /11.02.03.04/16.19.03 /34.11.03.07
44	3	350.10.10	2	2	-14.97	14.10/01.04.01/14.13/16.19.03/20.01.10/20.09.04
45	8	500.10.40	7	6	-38.44	12.04.01
47	4	410.40.30	5	3	-19.67	10.01.05.01/10.01.03.05/10.03.01.03/16.03.01 /18.02.01
48	3	410.40.90	3	3	-23.36	10.01.05.01/10.01.03.05
49	6	290.20.10	5	5	-35.34	20.09.04/20.01.10/14.04/42.16
50	4	550.1.29	16	4	-22.85	32.01/2.19/01.05.01.03.01/01.05.01.01.01
52	3	550.3.82	2	2	-14.97	12.04.03/01.03.16.01
53	13	260.30.20	11	6	-30.15	20.09.07.03
54	8	510.40.10	13	7	-40.63	11.02.03.01/11.02.03.04
58	5	550.2.436	2	2	-13.77	20.09.18
59	5	470.10	6	5	-35.34	01.04.01 /01.05.04/14.07.03/30.01.05 /43.01.03.05/11.02.03.04

TABLE 3: Continued.

Group ID	Group size	MIPS tag of protein complex	Protein complex size	Overlap (common proteins)	$\ln(P_{\text{overlap}})$	Main functional categories
61	5	440.12.10	7	5	-34.08	11.04.01/01.03.16.01
64	3	400	10	3	-18.57	43.01.03.09/01.06.01.07.11
66	5	550.1.212	35	5	-24.44	16.19.03/10.01.05.01/01.04.01/16.03.01
67	4	510.190.40	5	4	-28.75	11.04/01.03.16.01/11.02.03.04/34.11.03.07
70	3	440.12.30	3	3	-23.36	11.06.01/11.04.01
72	3	550.1.84	7	3	-19.80	20.09.13/16.01/14.10/20.09.18
74	3	510.180.30.10	2	2	-14.97	42.10.03/32.01.09/16.03.03/10.03.01.03 /10.01.09.05/10.01.05.01
76	7	290.10	9	7	-46.58	20.09.04/20.01.10/14.04/20.03
80	3	550.2.527	2	2	-14.97	30.01.05.01/18.02.01/18.01.01/14.07.03
81	3	260.90	6	3	-20.36	42.04.03/20.09.14/16.07/14.10/16.01/43.01.03.05
82	4	260.80	4	4	-30.35	42.25/20.09.13/20.09.07/20.09.07/20.09.16.09.03
86	4	510.180.20	7	4	-26.80	34.11.03.07/10.01.05.03.01/10.01.05.01/10.03.02
87	3	550.1.74	4	3	-21.97	20.09.07/42.04.03
89	3	550.2.321	6	3	-20.36	16.19.03 /14.13.01.01/14.07.11/01.04.01
91	3	550.2.317	3	3	-23.36	16.03.01/11.02.01/11.02.02
95	3	90.30	2	2	-14.97	34.01.01.03/14.10
96	4	110	4	4	-30.35	43.01.03.05/11.02.03.04/16.19.01/30.01.09.07 /14.07.03/01.04.01
97	3	260.30.30.20	2	2	-14.97	14.04 /20.09.07.05
100	12	140.20.30	7	4	-20.60	43.01.03.05/42.04/40.01
101	6	550.3.12	7	4	-24.09	10.03.01.01.03/43.01.03.05/30.01.05.01/14.07.03
102	9	445.20	4	4	-25.52	16.01/16.19.03/10.03.01.01.03/14.13.01.01/14.10 /14.07.05
103	7	140.20.20	25	5	-23.21	43.01.03.05/40.01/16.01/42.04.03/20.09.18.09.01
104	5	140.20.30	7	4	-25.19	43.01.03.05/42.01/32.01.03/20.09.18.09.01
105	11	260.60	10	10	-66.33	20.09.07.03
106	7	270.20.40	4	3	-18.42	10.03.04.03
108	9	133.10	10	7	-41.79	10.03.01.01/18.02.01
110	3	140.30.30	14	3	-17.46	20.09.14/10.03.05.01/10.03.04.09/10.03.01.01.11 /02.45.11
112	5	270.20.20	3	3	-21.05	10.03.04.05/16.01
113	5	270.20.10	3	3	-21.05	10.03.04.05

TABLE 4: Comparison of our method and CFinder in terms of the number of functional categories shared by all the proteins in the groups.

Functional categories common to all proteins in the group	Groups found by our program	Groups found by CFinder
< 1	9 (7.8%)	35 (18.0%)
= 1	19 (16.3%)	37 (18.9%)
= 2	27 (23.3%)	35 (18.0%)
> 2	61 (52.6%)	88 (45.1%)
Total	116 (100%)	195 (100%)

TABLE 5: Running times of the programs on 3 data sets of yeast protein interactions on a Pentium IV 3.0 GHz processor with 512 MB memory.

Program	MIPS data (4874 nodes 15660 edges)	DIP data (4932 nodes 17491 edges)	BOND data (18692 nodes 59516 edges)
Our program	10.06 s	13.06 s	1 m 06.63 s
CFinder	17.45 s	24.33 s	2 m 22.46 s

MIPS data: PPI_180105.tab from MIPS (1/18/2005).

DIP data: yeast20071104.lst from DIP (11/04/2007).

BOND data: data from BOND (11/9/2007).

complexes, which are cataloged in the MIPS *Saccharomyces cerevisiae* genome database.

ACKNOWLEDGMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) under Grant no. F01-2007-000-10140-0 and Grant no. F01-2007-000-10140-0 through the Systems Bio-Dynamics Research Center.

REFERENCES

- [1] D. Bu, Y. Zhao, L. Cai, et al., "Topological structure analysis of the protein-protein interaction network in budding yeast," *Nucleic Acids Research*, vol. 31, no. 9, pp. 2443–2450, 2003.
- [2] R. Battiti and M. Protasi, "Reactive local search for the maximum clique problem," *Algorithmica*, vol. 29, no. 4, pp. 610–637, 2001.
- [3] K. Katayama, A. Hamamoto, and H. Narihisa, "An effective local search for the maximum clique problem," *Information Processing Letters*, vol. 95, no. 5, pp. 503–511, 2005.
- [4] B. Adamcsek, G. Palla, I. Farkas, I. Derényi, and T. Vicsek, "CFinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, 2006.
- [5] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
- [6] B.-H. Ju and K. Han, "Complexity management in visualizing protein interaction networks," *Bioinformatics*, vol. 19, supplement 1, pp. i177–i179, 2003.
- [7] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Paris, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.
- [8] V. Spirin and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 12123–12126, 2003.
- [9] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 8, pp. 4569–4574, 2001.
- [10] P. Uetz, L. Giot, G. Cagney, et al., "A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*," *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.
- [11] A. Ruepp, A. Zollner, D. Maier, et al., "The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes," *Nucleic Acids Research*, vol. 32, no. 18, pp. 5539–5545, 2004.